UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/809,247	03/25/2004	Alan Tchochiev	MSFT122122	7234

26389 7590 05/17/2007
CHRISTENSEN, O'CONNOR, JOHNSON, KINDNESS, PLLC
1420 FIFTH AVENUE
SUITE 2800
SEATTLE, WA 98101-2347

EXAMINER	
WANG, BEN C	
ART UNIT	**PAPER NUMBER**
2192	
MAIL DATE	**DELIVERY MODE**
05/17/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/809,247	**Applicant(s)** TCHOCHIEV, ALAN	
	Examiner Ben C. Wang	**Art Unit** 2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *25 March 2004*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-56* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-56* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *31 October 2005* is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date *08/09/2004*.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

1. Claims 1-56 are pending in this application and presented for examination.

Claim Objections

2. Claims 14 and 37 are objected to because the following informalities:

- "customizing properties of the generator;", claim 14, line 4, should be corrected "customizing properties of the generator."
- "customizing properties of the generator;", claim 37, line 4, should be corrected "customizing properties of the generator."

Appropriate correction is required.

Claim Rejections – 35 USC § 101

3. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

4. Claims 1-6 are rejected under 35 U.S.C 101 because the claims are directed to non-statutory subject matter.

5. **In claims 1-6**, a "computer-readable medium" is being cited; however, it appears that the computer-readable medium would reasonably be interpreted by one of ordinary skill in the art as computer listings per se, are not physical

"things". They are neither computer components nor statutory processes, as they are not "act" being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer which permit the computer program's functionality to be realized. In contrast, a claimed computer readable medium encoded with a computer program is a computer element which defines structural and functional interrelationships between the computer program and the rest of the computer which permit the computer program's functionality to be realized, and is thus statutory. Accordingly, it is important to distinguish claims that define descriptive material per se from claims that define statutory inventions. (See MPEP 2106.01(I))

Claim Rejections – 35 USC § 102(b)

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102(b) that form the basis for the rejections under this section made in this office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6. Claims 1-11, 30-35, 53-54, and 55-56 are rejected under 35 U.S.C. 102(b) as being anticipated by M. K. Bowman-Amuah (Pat. No. US 6,442,748 B1) (hereinafter 'Bowman-Amuah')

7. **As to claim 1** (currently amended), Bowman-Amuah discloses A computer-readable medium having a base generator class stored thereon for use

by developers to create generators to perform specific tasks, the base generator class comprising: a base generator class constructor (i.e. Col. 285, Lines 21-42, class listing); a generator properties class that provides incrementation capability, which allows the value of a generator property to vary during consecutive executions of a generator (i.e. Col. 272, Line 66 through Col. 273, Line 6 – persistence is the capability to permanently store this data in its original or a modified state, until the information system purposely deletes); a status indicator (i.e. Col. 92, Lines 53-55; Col. 108, Lines 16-19; Col. 111, Line 37; Col. 112, Lines 2, 9-10, 41-43); a schedule class (i.e. Col. 74, Lines 21-23; Col. 100, Lines 20-24 – they provide services for scheduling, starting, stopping, and restarting both client and server tasks; Col. 108, Lines 30-34 – areas for design attention include scheduling…; Col. 109, Lines 40-44; Col. 118, Lines 47-49; Col. 188, Lines 43-46; Col. 194, Lines 1-2, 14-17); and a logging class (i.e. Fig. 12, environment --> application services --> logging; Col. 68., Lines 17-22; Col. 81, Lines 65-67; Col. 97, Lines 4-5; Col. 100, Lines 38-43; Col. 101, Lines 10-14 – logging services support the logging of informational, error, and warning messages; Col. 194, Lines 14-17, 56-60).

8. **As to claim 7** (original), Bowman-Amuah discloses a method of creating a generator, wherein the generator performs a specific task such as creating a file, comprising: creating a new generator class that inherits a base generator class that contains incrementation capability (i.e. Col. 11, Lines 36-58; Col. 12, Lines 3-8, 47-49; Col. 13, Lines 5-9 – sub-classing and inheritance make it possible to

extend and modify objects through deriving new kinds of objects from the standard classes available in the system; Col. 14, Lines 45-52); creating a public default constructor for the new generator class that overrides the base generator class constructor (i.e. Col. 11, Lines 55-58; Col. 14, Line 63 through Col. 15, Line 3 – to take full advantage a framework's reusable design, a programmer typically writes code that overrides and is called by the framework); and implementing a function in the new generator class to perform the specific task (i.e. Col. 14, Lines 24-26 – they build from there by replacing some of the generic capabilities of the framework with the specific capabilities of the intended application).

9. **As to claim 30** (currently amended), Bowman-Amuah discloses a method for object generation using a base generator class, comprising: creating a generator that performs a specific task (i.e. Col. 14, Lines 24-26 – they build from there by replacing some of the generic capabilities of the framework with the specific capabilities of the intended application); customizing the settings of the generator, the settings include including incrementation settings that specify how the value of a generator property may vary between generated objects (i.e. Col. 272, Line 66 through Col. 273, Line 6 – persistence is the capability to permanently store this data in its original or a modified state, until the information system purposely deletes); and executing the generator with the customized settings (i.e. Col. 186, Lines 8-20).

10. **As to claim 53** (original), Bowman-Amuah discloses a method of varying the value of a property associated with a task, during consecutive executions of the task, comprising: allowing the value of the property to vary during consecutive executions of the task (i.e. Col. 272, Line 66 through Col. 273, Line 6 – persistence is the capability to permanently store this data in its original or a modified state, until the information system purposely deletes); creating settings associated with the property that control how the value may vary during consecutive executions of the task (i.e. Col. 154, Lines 58-60; Col. 253, Lines 63-67; Col. 258, Lines 31-37; Col. 279, Line 65 through Col. 280, Line 2; Col. 299, Lines 32-35); and allowing a user executing the task to customize the settings according to user preference (Col. 252, Lines 49-51 – the predetermined criteria may include user preferences ...).

11. **As to claim 55** (original), Bowman-Amuah discloses a computer-readable medium containing computer-executable instructions for a method of varying the value of a property associated with a task, during consecutive executions of the task, the method comprising: allowing the value of the property to vary during consecutive executions of the task (i.e. Col. 272, Line 66 through Col. 273, Line 6 – persistence is the capability to permanently store this data in its original or a modified state, until the information system purposely deletes); creating settings associated with the property that control how the value may vary during consecutive executions of the task (i.e. Col. 154, Lines 58-60; Col. 253, Lines 63-67; Col. 258, Lines 31-37; Col. 279, Line 65 through Col. 280, Line 2; Col. 299,

Lines 32-35); and allowing a user executing the task to customize the settings according to user preference (Col. 252, Lines 49-51 – the predetermined criteria may include user preferences …).

12. **As to claim 2** (currently amended) (incorporating the rejection in claim 1), Bowman-Amuah discloses the computer-readable medium wherein the generator properties class that provides incrementation capability includes a plurality of generator properties (i.e. Fig. 58, steps 5802 – storing a plurality of attribute values …, 5804 – providing a plurality of attribute names in the attribute dictionary for the stored attribute values, 5806 – verifying that a current user is authorized to either set or get one of the attribute values upon a request which includes the attribute name that corresponds to the attribute value, 5808 – obtaining or updating the attribute value in the attribute dictionary if the verification is successful; Col. 272, Line 66 through Col. 273, Line 6 – persistence is the capability to permanently store this data in its original or a modified state, until the information system purposely deletes; Col. 137, Lines 16-20; Col. 139, Lines 54-56 – components can be designed so that their properties can be tailored to meet varying requirements;).

13. **As to claim 3** (original) (incorporating the rejection in claim 2), Bowman-Amuah discloses the computer-readable medium wherein said plurality of generator properties includes: a value of a generator property (Fig. 60, elements 6002 – setAttribute, 6000 - setAttribute); a plurality of incrementation settings (i.e.

Fig. 58, steps 5802 – storing a plurality of attribute values ..., 5804 – providing a plurality of attribute names in the attribute dictionary for the stored attribute values, 5806 – verifying that a current user is authorized to either set or get one of the attribute values upon a request which includes the attribute name that corresponds to the attribute value, 5808 – obtaining or updating the attribute value in the attribute dictionary if the verification is successful); a default incrementor that changes the value of the generator property (Col. 259, Lines 41-45; Col. 290, Lines 18-22); and a default validator that validates the value of the generator property (Fig. 129; Col. 250, Lines 9-22 – the framework would provide a common approach to validating user data across all of an application's user interfaces; while some common validation rules would be provided ...; Fig. 127, elements 12702 – validate(), 12706 – validate(); Fig. 131 – Validation Rule; Fig. 132; Col. 170, Lines 60-64).

14. **As to claim 4** (original) (incorporating the rejection in claim 1), Bowman-Amuah discloses the computer-readable medium wherein the status indicator includes a status user interface (UI) for displaying the execution status of generators (i.e. Fig. 129 – status; Col. 92, Lines 53-55; Col. 108, Lines 16-19; Col. 111, Line 37; Col. 112, Lines 2, 9-10, 41-43).

15. **As to claim 5** (currently amended) (incorporating the rejection in claim 1), Bowman-Amuah discloses the computer-readable medium wherein the schedule class comprises: a start condition under which the execution of a generator may

be started; a recurrence condition under which the execution of a generator may recur; an end condition under which the execution of a generator stops (i.e. Col. 51, Lines 8-16; Col. 74, Lines 21-23; Col. 100, Lines 20-24 – they provide services for scheduling, starting, stopping, and restarting both client and server tasks; Col. 108, Lines 30-34; Col. 109, Lines 36-37; Col. 113, Lines 26-29 – a set schedule and frequency); and a dialog box that can be used to accept user input (i.e. Fig. 164 – SAVE, CANCEL; Fig. 165; Fig. 39; Col. 130, Lines 23-25; Col. 14, Lines 18-26; Col. 34, Lines 62-66; Fig. 129).

16. **As to claim 6** (currently amended) (incorporating the rejection in claim 1), Bowman-Amuah discloses the computer-readable medium wherein the logging class enables the recording of the execution process of a generator (i.e. Fig. 12, environment --> application services --> logging; Col. 68., Lines 17-22; Col. 81, Lines 65-67; Col. 97, Lines 4-5; Col. 100, Lines 38-43; Col. 101, Lines 10-14 – logging services support the logging of informational, error, and warning messages; Col. 194, Lines 14-17, 56-60).

17. **As to claim 8** (original) (incorporating the rejection in claim 7) and **Claim 32** (original) (incorporating the rejection in claim 31), Bowman-Amuah discloses the method wherein creating a public default constructor comprises: initializing the base generator class constructor with the name and the description of the generator (Col. 204, Lines 22-52); and defining the properties of the generator (i.e. Fig. 58, steps 5802 – storing a plurality of attribute values ..., 5804 –

providing a plurality of attribute names in the attribute dictionary for the stored attribute values, 5806 – verifying that a current user is authorized to either set or get one of the attribute values upon a request which includes the attribute name that corresponds to the attribute value, 5808 – obtaining or updating the attribute value in the attribute dictionary if the verification is successful).

18. **As to claim 9** (original) (incorporating the rejection in claim 8) and **claim 33** (original) (incorporating the rejection in claim 32), Bowman-Amuah discloses the method wherein defining properties for the generator comprises: (a) defining the name of a property (Fig. 58, step 5804 – providing a plurality of attribute names ...); (b) setting a default value for the property (Col. 259, Lines 41-45; Col. 290, Lines 18-22); (c) providing a description for the property (Fig. 105, step 10506 – including a series of the attribute descriptors defining elements of the data); (d) specifying incrementation settings for the property (i.e. Fig. 58, steps 5802 – storing a plurality of attribute values ..., 5804 – providing a plurality of attribute names in the attribute dictionary for the stored attribute values, 5806 – verifying that a current user is authorized to either set or get one of the attribute values upon a request which includes the attribute name that corresponds to the attribute value, 5808 – obtaining or updating the attribute value in the attribute dictionary if the verification is successful); (e) creating a custom property incrementor, if applicable (Col. 14, Lines 18-26, 45-52); (f) creating a custom property validator, if applicable; (Fig. 129; Col. 250, Lines 9-22 – the framework would provide a common approach to validating user data across all of an

application's user interfaces; while some common validation rules would be provided ...; Fig. 127, elements 12702 – validate(), 12706 – validate(); Fig. 131 – Validation Rule; Fig. 132; Col. 170, Lines 60-64) and (g) repeating (a)-(f) for all properties of the generator (i.e. Col. 285, Lines 61-67).

19. **As to claim 10** (original) (incorporating the rejection in claim 7) and **claim 34** (original) (incorporating the rejection in claim 31), Bowman-Amuah discloses the method further comprising implementing a function to be executed before each execution of a generator (Fig. 136, step 13602 – a first assertion asserting a pre-condition ...; Col. 255, Lines 31-41 – having pre-conditions and post-conditions that must be satisfied for the operation to be successful).

20. **As to claim 11** (original) (incorporating the rejection in claim 7) and **claim 35** (original) (incorporating the rejection in claim 31), Bowman-Amuah discloses the method further comprising implementing a function to be executed after each execution of a generator (Fig. 136, step 13606 – a second assertion asserting a post-condition ...; Col. 255, Lines 31-41 – having pre-conditions and post-conditions that must be satisfied for the operation to be successful).

21. **As to claim 31** (original) (incorporating the rejection in claim 30), Bowman-Amuah discloses the method wherein creating a generator that performs a specific task comprises: creating a new generator class that inherits the base generator class (i.e. Col. 11, Lines 36-58; Col. 12, Lines 3-8, 47-49; Col.

13, Lines 5-9 – sub-classing and inheritance make it possible to extend and modify objects through deriving new kinds of objects from the standard classes available in the system; Col. 14, Lines 45-52); creating a public default constructor for the new generator class that overrides the base generator class constructor (i.e. Col. 11, Lines 55-58; Col. 14, Line 63 through Col. 15, Line 3 – to take full advantage a framework's reusable design, a programmer typically writes code that overrides and is called by the framework); and implementing a function in the new generator class to perform a specific task (i.e. Col. 14, Lines 24-26 – they build from there by replacing some of the generic capabilities of the framework with the specific capabilities of the intended application).

22. **As to claim 54** (original) (incorporating the rejection in claim 53), Bowman-Amuah discloses the method wherein the step of allowing the value of the property to vary during consecutive executions of the task further comprises: implementing a function that increments a property value according to the settings associated with the property that control how the value may vary during consecutive executions of the task (i.e. Col. 272, Line 66 through Col. 273, Line 6 – persistence is the capability to permanently store this data in its original or a modified state, until the information system purposely deletes).

23. **As to claim 56** (original) (incorporating the rejection in claim 55), Bowman-Amuah discloses the computer-readable medium wherein the step of allowing the value of the property to vary during consecutive executions of the

task further comprises: implementing a function that increments a property value according to the settings associated with the property that control how the value may vary during consecutive executions of the task (i.e. Col. 272, Line 66 through Col. 273, Line 6 – persistence is the capability to permanently store this data in its original or a modified state, until the information system purposely deletes).

24. Claims 12-14, 16-17, 20, 28, 36-37, 39-40, 43, 51 are rejected under 35 U.S.C. 102(b) as being anticipated by L. M. Moore (Pub. No. US 2002/0078069 A1) (hereinafter 'Moore')

25. **As to claim 12** (original), Moore discloses a method of using a generator that performs a specific task such as creating a file (i.e. [0008] – to provide a mechanism for automatically generating file names .... To include some defining characteristic ...), comprising: customizing the settings of a generator (i.e. [0008]), the settings include including incrementation settings that specify how the value of a generator property may vary between generated objects (i.e. [0012], Lines 4-16 – the argument may require calculation of a time and/or date or of an incremental number, or require that the user be prompted for input; [0023] – the generated text may be, for example, the current date, the current time, a user input, and/or an incremental numerical value; [0031]; P. 4, Left-Col, Line 35 – calculating an incremental value); and executing the generator with the customized settings (i.e. [0012], Lines 10-16 – the new document may be saved

by concatenating text string(s) for the fixed portion(s) and the calculated or received text string(s) for the variable portions).

26. **As to claim 13** (original) (incorporating the rejection in claim 12) and **claim 36** (original) (incorporating the rejection in claim 30), Moore discloses the method customizing the settings of a generator, is accomplished through a user interface (Fig. 2; [0016]; [0022]; [0024]; [0026]; [0039]).

27. **As to claim 14** (original) (incorporating the rejection in claim 13) and **claim 37** (original) (incorporating the rejection in claim 36), Moore discloses the method further comprising: starting an object generator user interface; selecting a generator; and customizing properties of the generator (Fig. 2; [0016]; [0022]; [0024]; [0026]; [0039]).

28. **As to claim 16** (original) (incorporating the rejection in claim 14) and **claim 39** (original) (incorporating the rejection in claim 37), Moore discloses the method further comprising loading the settings of a generator from a file Fig. 4B; [0046], Lines 10-13.

29. **As to claim 17** (original) (incorporating the rejection in claim 14) and **claim 40** (original) (incorporating the rejection in claim 37), Moore discloses the method wherein customizing the properties of the generator comprises: (a) selecting a property (Fig. 2); (b) specifying the value of the property; (c)

specifying the incrementation settings of the property (i.e. [0012], Lines 4-16 – the argument may require calculation of a time and/or date or of an incremental number, or require that the user be prompted for input; [0023] – the generated text may be, for example, the current date, the current time, a user input, and/or an incremental numerical value; [0031]; P. 4, Left-Col, Line 35 – calculating an incremental value); and (d) repeating (a)-(c) until there are no more properties to be customized.

30. **As to claim 20** (original) (incorporating the rejection in claim 14) and **claim 43** (original) (incorporating the rejection in claim 37), Moore discloses the method further comprising saving the settings of the generator (Fig. 4B; [0046], Lines 10-13).

31. **As to claim 28** (original) (incorporating the rejection in claim 12) and **claim 51** (original) (incorporating the rejection in claim 30), Moore discloses the method further comprising executing the generator through a user interface (Fig. 2; [0022).

Claim Rejections – 35 USC § 103(a)

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary

> skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

32. Claims 15, 18-19, 21-27, 29, 38, 41-42, 44-50, and 52 are rejected under 35 U.S.C. 103(a) as being unpatentable over Moore in view of Bowman-Amuah.

33. **As to claim 15** (original) (incorporating the rejection in claim 14) and **claim 38** (original) (incorporating the rejection in claim 37), Moore does not explicitly disclose the method selecting a generator further comprising adding a generator from files containing one or more generators.

However, in an analogous art of system of method and article of manufacture for a persistent state and persistent object separator in an information services patterns environment, Bowman-Amuah discloses the method selecting a generator further comprising adding a generator from files containing one or more generators (i.e. Col. 71, Lines 18-20; Col. 99, Lines 1-4).

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Bowman-Amuah into the Moore's system to provide the method selecting a generator further comprising adding a generator from files containing one or more generators in Moore system.

The motivation is that it would further enhance the Moore's system by taking, advancing and/or incorporating Bowman-Amuah's system which offers significant advantages that persistence is the capability to permanently store this data in its original or a modified state and hence provides incrementation capability for a generator properties as once suggested by Bowman-Amuah (i.e. Col. 272, Line 66 through Col. 273, Line 6).

34. **As to claim 18** (original) (incorporating the rejection in claim 14) and **claim 41** (original) (incorporating the rejection in claim 37), Moore does not explicitly disclose the method further comprising setting a schedule for executing the generator.

However, in an analogous art of system of method and article of manufacture for a persistent state and persistent object separator in an information services patterns environment, Bowman-Amuah discloses the method further comprising setting a schedule for executing the generator (i.e. Col. 74, Lines 21-23; Col. 100, Lines 20-24 – they provide services for scheduling, starting, stopping, and restarting both client and server tasks; Col. 108, Lines 30-34 – areas for design attention include scheduling…; Col. 109, Lines 40-44; Col. 118, Lines 47-49; Col. 188, Lines 43-46; Col. 194, Lines 1-2, 14-17).

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Bowman-Amuah into the Moore's system to provide the method further comprising setting a schedule for executing the generator in Moore system.

The motivation is that it would further enhance the Moore's system by taking, advancing and/or incorporating Bowman-Amuah's system which offers significant advantages that persistence is the capability to permanently store this data in its original or a modified state and hence provides incrementation capability for a generator properties as once suggested by Bowman-Amuah (i.e. Col. 272, Line 66 through Col. 273, Line 6).

35. **As to claim 19** (original) (incorporating the rejection in claim 14) and **claim 42** (original) (incorporating the rejection in claim 37), Moore does not explicitly disclose the method further comprising setting logging options for executing the generator.

However, in an analogous art of system of method and article of manufacture for a persistent state and persistent object separator in an information services patterns environment, Bowman-Amuah discloses the method further comprising setting logging options for executing the generator (i.e. Fig. 12, environment --> application services --> logging; Col. 68., Lines 17-22; Col. 81, Lines 65-67; Col. 97, Lines 4-5; Col. 100, Lines 38-43; Col. 101, Lines 10-14 – logging services support the logging of informational, error, and warning messages; Col. 194, Lines 14-17, 56-60).

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Bowman-Amuah into the Moore's system to provide the method further comprising setting logging options for executing the generator in Moore system.

The motivation is that it would further enhance the Moore's system by taking, advancing and/or incorporating Bowman-Amuah's system which offers significant advantages that persistence is the capability to permanently store this data in its original or a modified state and hence provides incrementation capability for a generator properties as once suggested by Bowman-Amuah (i.e. Col. 272, Line 66 through Col. 273, Line 6).

36. **As to claim 21** (original) (incorporating the rejection in claim 12) and **claim 44** (original) (incorporating the rejection in claim 30), Moore discloses the method customizing the settings of a generator, is accomplished programmatically.

However, in an analogous art of system of method and article of manufacture for a persistent state and persistent object separator in an information services patterns environment, Bowman-Amuah discloses the method customizing the settings of a generator, is accomplished programmatically (i.e. Col. 35, Lines 17-29; Col. 71, Lines 13-20; Col. 95, Lines 12-15).

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Bowman-Amuah into the Moore's system to provide the method customizing the settings of a generator, is accomplished programmatically in Moore system.

The motivation is that it would further enhance the Moore's system by taking, advancing and/or incorporating Bowman-Amuah's system which offers significant advantages that persistence is the capability to permanently store this data in its original or a modified state and hence provides incrementation capability for a generator properties as once suggested by Bowman-Amuah (i.e. Col. 272, Line 66 through Col. 273, Line 6).

37. **As to claim 22** (original) (incorporating the rejection in claim 21) and **claim 45** (original) (incorporating the rejection in claim 44), Bowman-Amuah discloses the method further comprising: creating a new instance of the generator; setting the number of objects to be generated by the generator (i.e. Col. 192, Lines 39-43; Col. 211, Lines 25-28; Col. 240, Lines 8-11; Col. 241, Lines 33-36); and customizing the properties of the generator (i.e. Col. 186, Lines 8-20).

38. **As to claim 23** (original) (incorporating the rejection in claim 22) and **claim 46** (Original) (incorporating the rejection in claim 45), Moore discloses the method wherein customizing the properties of the generator comprises: (a) setting the value of a property; (b) specifying the incrementation settings of the property and (c) repeating (a)-(b) until there are no more properties to be customized ([0012], Lines 8-10 – the argument may require calculation of a time and /or date or of an incremental number, or require that the user be prompted for input).

39. **As to claim 24** (original) (incorporating the rejection in claim 21) and **claim 47** (original) (incorporating the rejection in claim 44), Moore discloses the method further comprising: creating a new instance of the generator (i.e. Col.

192, Lines 39-43; Col. 211, Lines 25-28; Col. 240, Lines 8-11; Col. 241, Lines 33-36); and loading saved settings of the generator from a file (Fig. 4B; [0046], Lines 10-13).

40. **As to claim 25** (original) (incorporating the rejection in claim 21) and **claim 48** (original) (incorporating the rejection in claim 44), Moore discloses the method further comprising: creating a new instance of the generator (i.e. Col. 192, Lines 39-43; Col. 211, Lines 25-28; Col. 240, Lines 8-11; Col. 241, Lines 33-36); loading saved settings of the generator from a file (Fig. 4B; [0046], Lines 10-13); and Bowman-Amuah discloses implementing a function to execute the generator asynchronously (i.e. Col. 51, Lines 37-44; Col. 67, Lines 13-20; Col. 70, Lines 49-54).

41. **As to claim 26** (original) (incorporating the rejection in claim 21) and **claim 49** (original) (incorporating the rejection in claim 44), Bowman-Amuah discloses the method further comprising: creating a new instance of the generator (i.e. Col. 192, Lines 39-43; Col. 211, Lines 25-28; Col. 240, Lines 8-11; Col. 241, Lines 33-36); displaying an object generation status UI; and adding the generator to the object generation status UI (i.e. Fig. 129 – status; Col. 92, Lines 53-55; Col. 108, Lines 16-19; Col. 111, Line 37; Col. 112, Lines 2, 9-10, 41-43); Moore discloses loading saved settings of the generator from a file (Fig. 4B; [0046], Lines 10-13).

42. **As to claim 27** (original) (incorporating the rejection in claim 21) and **claim 50** (original) (incorporating the rejection in claim 44), Moore discloses loading saved settings of the generator from a file (Fig. 4B; [0046], Lines 10-13). Bowman-Amuah discloses the method further comprising: creating a new instance of the generator (i.e. Col. 192, Lines 39-43; Col. 211, Lines 25-28; Col. 240, Lines 8-11; Col. 241, Lines 33-36); displaying a schedule dialog box that allows a user to specify a schedule for executing the generator (i.e. Col. 74, Lines 21-23; Col. 100, Lines 20-24 – they provide services for scheduling, starting, stopping, and restarting both client and server tasks; Col. 108, Lines 30-34 – areas for design attention include scheduling...; Col. 109, Lines 40-44; Col. 118, Lines 47-49; Col. 188, Lines 43-46; Col. 194, Lines 1-2, 14-17); and displaying a logging dialog box that allows a user to specify logging options for executing the generator (i.e. Fig. 12, environment --> application services --> logging; Col. 68., Lines 17-22; Col. 81, Lines 65-67; Col. 97, Lines 4-5; Col. 100, Lines 38-43; Col. 101, Lines 10-14 – logging services support the logging of informational, error, and warning messages; Col. 194, Lines 14-17, 56-60).

43. **As to claim 29** (original) (incorporating the rejection in claim 12) and **claim 52** (original) (incorporating the rejection in claim 30), Moore discloses executing the generator through a user interface, but does not explicitly disclose the method further comprising executing the generator programmatically.

However, in an analogous art of system of method and article of manufacture for a persistent state and persistent object separator in an information services patterns environment, Bowman-Amuah discloses the method further comprising executing the generator programmatically (i.e. Col. 35, Lines 17-29; Col. 71, Lines 13-20; Col. 95, Lines 12-15).

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Bowman-Amuah into the Moore's system to further provide the method further comprising executing the generator programmatically in Moore system.

The motivation is that it would further enhance the Moore's system by taking, advancing and/or incorporating Bowman-Amuah's system which offers significant advantages that persistence is the capability to permanently store this data in its original or a modified state and hence provides incrementation capability for a generator properties as once suggested by Bowman-Amuah (i.e. Col. 272, Line 66 through Col. 273, Line 6).

Conclusion

44. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- Nachef et al., Dynamic Creation of Object Classes (Pat. No. US 7,143,416 B1)
- Henninger et al., Method and Apparatus for Automatic Generation of Object Oriented Code for Mapping Relational Data to Objects (Pat. No. 5,499,371)
- M. K. Bowman-Amuah, Attribute Dictionary in a Business Logic Services Environment (Pat. No. US 6,601,234 B1)
- M. K. Bowman-Amuah, Piecemeal Retrieval in an Information Services Patterns Environment (Pat. No. US 6,550,057 B1)
- Iborra et al., Automatic Software Production System (Pub. No. 2002/0062475 A1)

45. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ben C. Wang whose telephone number is 571-270-1240. The examiner can normally be reached on Monday - Friday, 8:00 a.m. - 5:00 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

TUAN DAM
SUPERVISORY PATENT EXAMINER

BCW

May 7, 2007